IEEE 802.1 Security MACsec and MAC Privacy YANG Some Recent Updates

Don Fedyk – don.Fedyk@labn.net

Disclaimer

- This is a work in progress. The material here is for discussion purposes and may contain errors.

Revised Prototype YANG Model (snippet)

```
module: ieee802-dot1ae-pry

augment /if:interfaces/if:interface:
 +--rw pry {macsec-priv}?
    +--rw mac-privacy-enabled?          boolean
    +--rw pry-source-address?           Union
    +--rw pry-destination-address?      union
    +--rw user-priority-to-pry* [user-priority]
    |  +--rw user-priority    uint8
    |  +--rw privacy-type?    union
    +--rw privacy-channel* [channel-id]
    |  +--rw channel-id                identityref
    |  +--rw user-data-frame-size?     uint16
    |  +--rw mppdu-priority?           dot1q-types:priority-type
    |  +--rw requested-bit-rate?       uint64
    |  +--ro actual-bit-rate?          uint64
    |  +--rw burst-size?               uint32
    |  +--ro total-size-on-wire?       uint16
    |  +--rw fragment-enable?          boolean
    +--rw privacy-frame* [frame-id]
    |  +--rw frame-id                  identityref
    |  +--rw user-data-frame-size?     uint16
    |  +--rw mppdu-priority?           dot1q-types:priority-type
    |  +--rw nearest-multiple-pad?     uint16augment
```

What gets configured

What gets used – System may adjust

A default for PAE address using a Union

```
leaf pry-source-address {
  type  union{
    type ieee:mac-address;
    type string {
         pattern “([Pp][Aa][Ee] [Aa]ddress)";
    }
  }
  default "PAE address";
  description
    "The individual MAC address of the MAC Privacy service. This
    MAC address may be shared with other components. By setting
     it to PAE address it is the Nearest non-TPMR Bridge group
     address, 01-80-C2-00-00-03 shared with PAE";
  reference
    "IEEE 801.1AE Clause  20.13.2";
}
```

This adds an option to match the PAE address

Union allows any MAC address & “PAE Address” or “pae address”
Limited strict match combinations
01-80-C2-00-00-03 is the PAE address and this works too

A default for PAE address using an additional configuration parameter

```
      leaf use-pae-address {
          type boolean;
          config true;
          default true;
          description
            "By setting to PAE address true it is the Nearest non-TPMR
             Bridge group address, 01-80-C2-00-00-03 shared with PAE
             for both source and destination address. This value overrides
             any values in source and destination address when true.";
          reference
            "IEEE 801.1AE Clause  20.13.6.7";
      }
```

```
Align with MIB style
Now this variable overrides both source and dest address if they are set.
```

Functionally both options are the same .

You should align source /destination config in this case

Fragmentation Enable

```
  leaf fragment-enable {
    type boolean;
    config true;
    default true;
    description
      "Fragmentation control for this channel. It is recommended
      to use fragmentation at all times for efficiency and minimizing
      delay. This control allows for showing the effects of fragmentation
      vs no fragmentation or simple performance tests.";
    reference
      "IEEE 801.1AE Clause  20.13.6.7";
  }
}
```

Yanglint Validation

```
"ieee802-dot1ae-pry:pry": {
     "pry-source-address": "11-22-33-44-55-66",
     "pry-destination-address": "PAE address",
     "user-priority-to-pry": [
      {
       "user-priority": 0,
       "privacy-type": "none"
      },
      {
       "user-priority": 1,
       "privacy-type": "frame-id-a"
      },
      {
       "user-priority": 2,
       "privacy-type": "express-channel"
      },
      {
       "user-priority": 3,
       "privacy-type": "express-channel"
      },
      {
       "user-priority": 4,
       "privacy-type": "preemptable-channel"
      },
      {
       "user-priority": 5,
       "privacy-type": "preemptable-channel"
      },
      {
       "user-priority": 6,
       "privacy-type": "preemptable-channel"
      },
      {
       "user-priority": 7,
       "privacy-type": "preemptable-channel"
      }
     ],
```

```
"privacy-channel": [
      {
       "channel-id": "preemptable-channel",
       "user-data-frame-size": 1518,
       "mppdu-priority": 3,
       "requested-bit-rate": "10000000000",
       "actual-bit-rate": "9705882352",
       "total-size-on-wire": 1564,
       "burst-size": 10000
       "fragment-enable": true

      }
     ],
     "privacy-frame": [
      {
       "frame-id": "frame-id-a",
       "user-data-frame-size": 1518,
       "mppdu-priority": 6,
       "nearest-multiple-pad": 16
      }
     ]
    },
```

Secy Traffic and SecY Access Priority

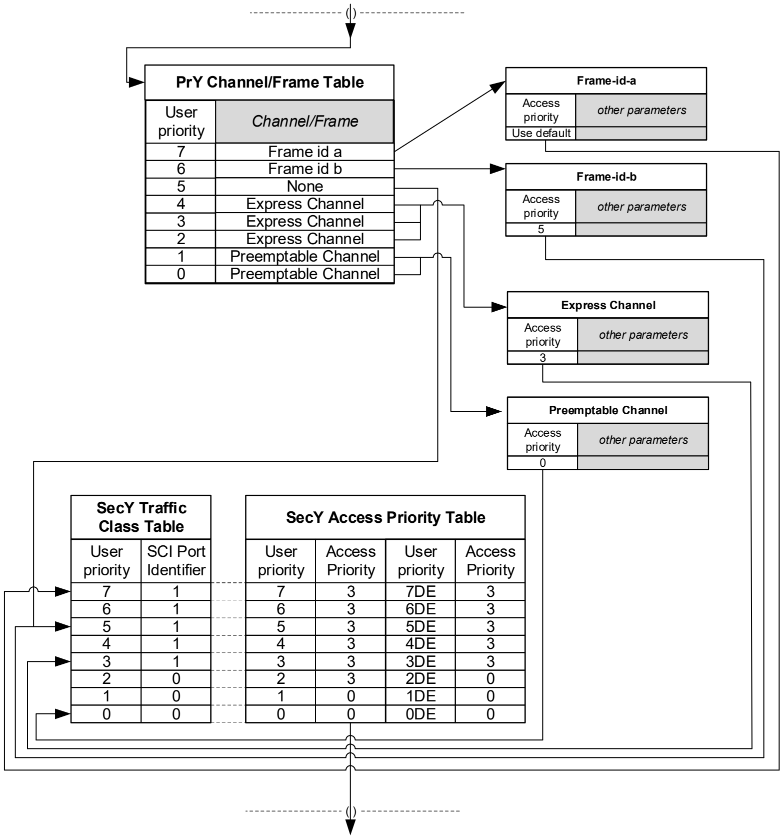
Figure 17-3—Priority handling and channel assignment

```
list user-priority-tc {
          key "user-priority";
          description
            "Each entry in the Traffic Class Table is a traffic class,
            represented by an integer from 0 (default) through 7 that also
            comprises the numeric value of the four most significant bits
            of the Port Identifier component of the SCI for the selected
            SC. The default for this table is every row has a non-mapping
            priority with the first row having all zeros, the second row
            having all ones etc. up to the last row having all sevens.";
          reference
            "IEEE 802.1AE-2018 Clause 10.7.17";
          leaf user-priority {
            type dot1q-types:priority-type;
            description
              "The User Priority";
            reference
              "IEEE 802.1AE-2018 Clause 10.7.17";
          }
          leaf traffic-class {
            type dot1q-types:priority-type;
            description
              "The traffic class that maps to the four most significant
              bits of the Port Identifier component of the SCI for the
              selected SC";
            reference
              "IEEE 802.1AE-2018 Clause 10.7.17";
          }
          leaf access-class-de0 {
            type uint8 {
              range "0..15";
            }
            description
              "The Access priority when PCP Discard eligible is not
              set(0). Access Priority is the high 3 bits and the
               DE bit is the lower bit. ";
            reference
              "IEEE 802.1AE-2018 Clause 10.7.17";
          }
```

```
leaf access-class-de1 {
            type uint8 {
              range "0..15";
          }
          description
            "The Access priority when PCP Discard eligible is
             set(1). Access Priority is the high 3 bits and the
             DE bit is the lower bit. ";
          reference
            "IEEE 802.1AE-2018 Clause 10.7.17";
          }
        }
```

This can be broken into two list one
For SecY traffic Class and one for Access
Priority but see next page.

Diagram that corresponds to the YANG



PrY Channel/Frame Table

User priority	*Channel/Frame*
7	Frame id a
6	Frame id b
5	None
4	Express Channel
3	Express Channel
2	Express Channel
1	Preemptable Channel
0	Preemptable Channel

Frame-id-a

Access priority	*other parameters*
Use default	

Frame-id-b

Access priority	*other parameters*
5	

Express Channel

Access priority	*other parameters*
3	

Preemptable Channel

Access priority	*other parameters*
0	

SecY Traffic Class Table

User priority	SCI Port Identifier
7	1
6	1
5	1
4	1
3	1
2	0
1	0
0	0

SecY Access Priority Table

User priority	Access Priority	User priority	Access Priority
7	3	7DE	3
6	3	6DE	3
5	3	5DE	3
4	3	4DE	3
3	3	3DE	3
2	3	2DE	0
1	0	1DE	0
0	0	0DE	0

Yanglint Validation

```
"ieee802-dot1ae:secy": {
         "controlled-port-number": 1,
         "verification": {
           "validate-frames": "strict",
           "replay-protect": true
         },
         "generation": {
           "max-transmit-channels": 16,
           "max-transmit-keys": 16,
           "protect-frames": true,
           "always-include-sci": true,
           "use-es": true,
           "use-scb": true,
           "user-priority-tc": [
             {
               "user-priority": 0,
               "traffic-class": 0,
               "access-class-de0": 0,
               "access-class-de1": 0
             },
             {
               "user-priority": 1,
               "traffic-class": 1
             },
             {
               "user-priority": 2,
               "access-class-de0": 2,
               "access-class-de1": 2
             },
             {
               "user-priority": 3,
               "traffic-class": 3,
               "access-class-de0": 3,
               "access-class-de1": 3
             }
           ]
         }
       },
```

While it is one list with multiple rows
Configuration can group it by configuration

SecY Traffic Traffic Class

SecY Access Priority

Both together

While I have your attention

- YANG cannot default the values previous list. But the backend code can do this. <- So, no different to the user. We specify the default table population in Standard text.
- There is a way to rearrange the list to a set of containers with names that then specifies YANG defaults.
- This blows up the code by 7 – 15 times and adds no real value – it actually makes the YANG harder to read. I had rejected this format of coding – but some projects have used this in the past. When I see this, I comment on it, but the projects were before I was reviewing them.

Comments?
Thank You